



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Jiří Berný

Smart plant pot

Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: doc. RNDr. Tomáš Bureš, Ph.D.

Studijní program: Informatika

Studijní obor: Softwarové a datové inženýrství

Praha 2018

Děkuji panu doc. RNDr. Tomášovi Burešovi, Ph.D a RNDr. Dominikovi Škodovi za
zapůjčení hardwaru a nápomoc při vypracování.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 18. 8. 2018

Jiří Berný

Název práce: Smart plant pot

Autor: Jiří Berný

Katedra / Ústav: Katedra distribuovaných a spolehlivých systémů

Vedoucí bakalářské práce: doc. RNDr. Tomáš Bureš, Ph.D., Katedra distribuovaných a spolehlivých systémů

Abstrakt: Spousta chytrých zařízení ulehčuje lidem život. Chytrý květináč umožní pěstitelům kontrolovat životní podmínky rostlin. Pomocí mikropočítače STM32F4 a webové aplikace může mít uživatel své květiny pod kontrolou odkudkoli i z mobilního telefonu. Mikropočítač si uživatel umístí ke své rostlině, připojí ho k WiFi síti a pomocí webová aplikace pozoruje poslední naměřená data, grafy s dlouhodobým vývojem prostředí a spravuje seznam svých květináčů. Backend webové aplikace je postaven na rozhraní OWIN a WebAPI. Frontend má podobu Single Page Application s použitím moderního frameworku Angular 5.

Klíčová slova: chytrý květináč, Internet věcí

Title: Smart plant pot

Author: Jiří Berný

Department: Department of Distributed and Dependable Systems

Supervisor: doc. RNDr. Tomáš Bureš, Ph.D., Department of Distributed and Dependable Systems

Abstract: Many of the clever devices are making peoples lifes easier. A Smart plant pot will anable gardeners to control the life environment of plants. Using the STM32F4 microcomputer and the web aplication, the user has all of his plants under control from any place even from his own mobile phone. The user places the microcomputer to his plant pot, connects it into his WiFi network and checks the last-measured data, graphs with long-term environment charts and manages his list of pots. Web applications backend is built on OWIN and WebAPI. Frontend was made with modern framework Angular 5 and has a form of a Single Page Application.

Keywords: smart plant pot, Internet of Things

Obsah

1. Úvod	1
2. Cíle projektu	3
3. Požadavky na řešení	4
3.1. Požadavky na hardware	4
3.2. Přenositelnost	4
3.3. Požadavky výkon	4
3.4. Návrh uživatelského rozhraní	4
3.4.1. Menu	5
3.4.2. Nastavení	6
3.4.3. Výsledky měření	7
4. Analýza řešení	8
5. Výběr hardwaru	10
6. Technologie	12
6.1. Měřicí zařízení	12
6.2. Konfigurační aplikace	12
6.3. Webová aplikace na straně serveru	12
6.4. Webová aplikace na straně klienta	12
7. Řešení	14
7.1. Firmware měřicího zařízení	14
7.2. Aplikace pro konfiguraci	14
7.3. Serverová část webové aplikace	14
7.4. Klientská část webové aplikace	15
8. Detaily implementace	17
8.1. Firmware měřicího zařízení	17
8.1.1. Komunikace se senzorem SHT1x	17
8.1.2. Komunikace s WiFi Plus click	17
8.1.3. Komunikační protokol mezi měřícím zařízením a serverem	18
8.1.4. Komunikační protokol mezi konfigurační aplikací a měřícím zařízením	18
8.2. Webová aplikace na straně serveru	23
8.2.1. MeasureCatcher	23
8.2.2. SyncGarden	23
8.2.3. ActivePot	23
8.2.4. Model	23
8.2.5. WebAPI	24

8.3.	Klientský kód webové aplikace	25
8.3.1.	AppModule.....	25
8.3.2.	AppComponent.....	25
8.3.3.	LoginComponent a RegisterComponent.....	25
8.3.4.	GardenComponent.....	25
8.3.5.	SettingsModal	25
8.3.6.	ChartComponent.....	25
8.3.7.	PotService.....	26
9.	Validace a zhodnocení	27
9.1.	Testování.....	27
9.2.	Obsluha více uživatelů	27
9.3.	Uživatelské rozhraní.....	28
9.3.1.	Menu	28
9.3.2.	Nastavení.....	29
9.3.3.	Grafy.....	30
10.	Související projekty	32
10.1.	Parrot Pot.....	32
10.2.	Planty	32
10.3.	DIY	32
10.4.	Porovnání.....	32
11.	Závěr.....	34
	Seznam použité literatury.....	35
	Apendix 1 – kompilace a konfigurace	38
	Měřicí zařízení.....	38
	Webová aplikace-server.....	38
	Kompilace.....	38
	Konfigurace	38
	Nastavení certifikátu	38
	Inicializace	39
	Webová aplikace-klient.....	39
	Apendix 2 – Uživatelská dokumentace	40
	Spuštění měřicího zařízení	40
	Webová aplikace-server.....	41
	Nastavení vlastní URL	41
	Webová aplikace-klient.....	41
	Přihlášení uživatele	41

Registrace nového uživatele.....	41
Základní práce s květináči.....	41
Nastavení květináče	42
Prohlížení grafů.....	42
Apendix 3 – přiložené CD	43

1. Úvod

Pojem Internet věcí byl poprvé použit v roce 1999 britským technologickým průkopníkem Kevinem Ashtonem pro označení systému, který pomocí senzorů umístí objekty z reálného světa do světa Internetu. Mezi populární implementace se řadí bezpečnostní systémy budov, systémy pro úsporu energie domácností nebo například autonomní vozidla [1]. Internet věcí je poměrně nové odvětví, které je čím dál více populární.

“Neptejte se, co bude v budoucnosti Internetu věcí

Ptejte se, co se děje v přítomnosti. Každý den jsou miliony zařízení, a data s nimi svázaná, přidávány do směsi Internetu. Pokud tato data nekontrolujete a nepracujete s nimi ve prospěch vás a vašich zákazníků, pak přicházíte o miliony příležitostí.“ [2]

Internet věcí je na vzestupu a stále více jeho zařízení ulehčuje lidem práci. Toto odvětví s sebou přináší několik problémů. Kromě požadavků na zabezpečení uživatelských dat je nutné řešit návrh samotného zařízení, jeho identifikaci, komunikaci, a zpracování a prezentování dat. Dalším důležitým faktorem je cena chytrého zařízení, kterou musí vyvážit jeho přidaná hodnota. Tato práce popisuje implementaci jednoho takového zařízení (chytrého květináče), manipulaci s daty a jejich prezentaci.

Chytrý květináč umožní pěstitelům monitorovat životní prostředí rostlin. Díky němu mohou pěstitelé sledovat stav a případné výkyvy prostředí rostlin a lépe reagovat na jeho změny. Práce se zabývá komunikací mikropočítače se senzory a jeho připojení do WiFi sítě, zpracováním naměřených hodnot na straně serveru a prezentací dat uživateli.

V následující kapitole jsou popsány cíle projektu. V kapitole č. 3 se nacházejí požadavky na zvolené řešení. Analýzou řešení se zabývá kapitola č. 4. Následují kapitoly s výběrem hardwaru pro měřicí zařízení a výběr použitých technologií. V kapitole č. 7 je stručný popis řešení. Detaily implementace se nacházejí v kapitole č. 8. Devátá kapitola se věnuje validaci a zhodnocení. Popisuje testování a výsledné

uživatelské rozhraní. V kapitole 10 je poukázáno na existující podobné projekty a porovnání s touto prací. Výsledek projektu shrnuje jedenáctá kapitola. Následuje přehled použité literatury. Apendix 1 je věnován postupu pro konfiguraci a kompilaci. Apendix 2 popisuje uživatelskou dokumentaci. Obsah přiloženého CD popisuje Apendix 3.

2. Cíle projektu

Cílem projektu je navrhnout systém pro chytré květináče v podobě mikropočítače se senzory, který umožní pěstitelům měřit a sledovat prostředí rostliny. Uživatel si takové zařízení umístí v blízkosti rostliny a připojí ho ke své WiFi síti. Zařízení začne odesílat naměřené hodnoty na server. Uživatel následně může pozorovat a kontrolovat přijatelnost prostředí rostliny prostřednictvím webové aplikace.

Každý uživatel může vlastnit libovolné množství těchto zařízení, proto je dobré brát v potaz snadnou správu a prohlížení naměřených hodnot počítající nejen s jedním zařízením. Zároveň lze předpokládat, že uživatele nebudou zajímat jen aktuální hodnoty, ale hlavně, jak se měřené prostředí mění v čase.

Řešení tohoto projektu bude obsahovat výběr vhodného hardwaru a implementaci jeho firmwaru pro měřící zařízení, a návrh aplikace pro server pro zpracování naměřených dat a interakci s uživateli. Uživatelé budou mít možnost pozorovat aktuální naměřené hodnoty, prohlížet grafy za zvolené časové období, přidávat, odebírat a pojmenovávat květináče.

Cílem projektu není implementace bezpečnostních mechanismů. Bude použito zabezpečení, které implementují použité technologie. Pokud žádné řešení nebude dostupné, bude pouze upozorněno na potenciální slabinu v zabezpečení.

3. Požadavky na řešení

3.1. Požadavky na hardware

Zvolený hardware pro měřicí zařízení by měl splňovat nízkou pořizovací cenu a energetické nároky. Zároveň by zařízení mělo být jednoduché na sestavení a uživatelskou obsluhu. Pro účely této práce budou použity jen senzory pro teplotu a vlhkost, nicméně projekt nebude neslučitelný s dalšími rozšířeními.

3.2. Přenositelnost

Řešení musí umožnit manipulaci s květináči a prezentaci dat provádět na mobilních i desktopových zařízeních s běžnými operačními systémy. Konfigurační aplikace měřicího zařízení bude přenositelná na úrovni zdrojového kódu mezi současnými desktopovými platformami. Přenositelnost je důležitá především na straně klienta. Na straně serveru je přenositelnost méně podstatná.

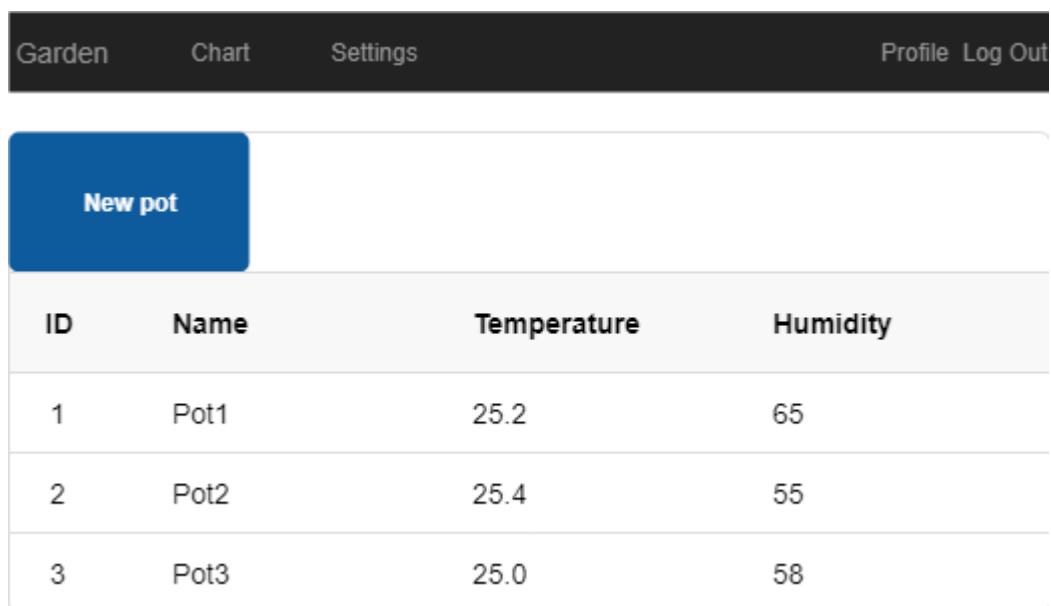
3.3. Požadavky výkon

Server by měl být schopen obsluhovat minimálně stovky uživatelů najednou. Na straně klienta je důležité, aby ovládání působilo maximálně plynule.

3.4. Návrh uživatelského rozhraní

Následuje ukázka rozhraní na straně klienta. Jsou zde použity nákresy pro ilustraci toho, jak bude vypadat komunikace s uživatelem. Výsledný vzhled a ovládací prvky budou upraveny dle potřeby.

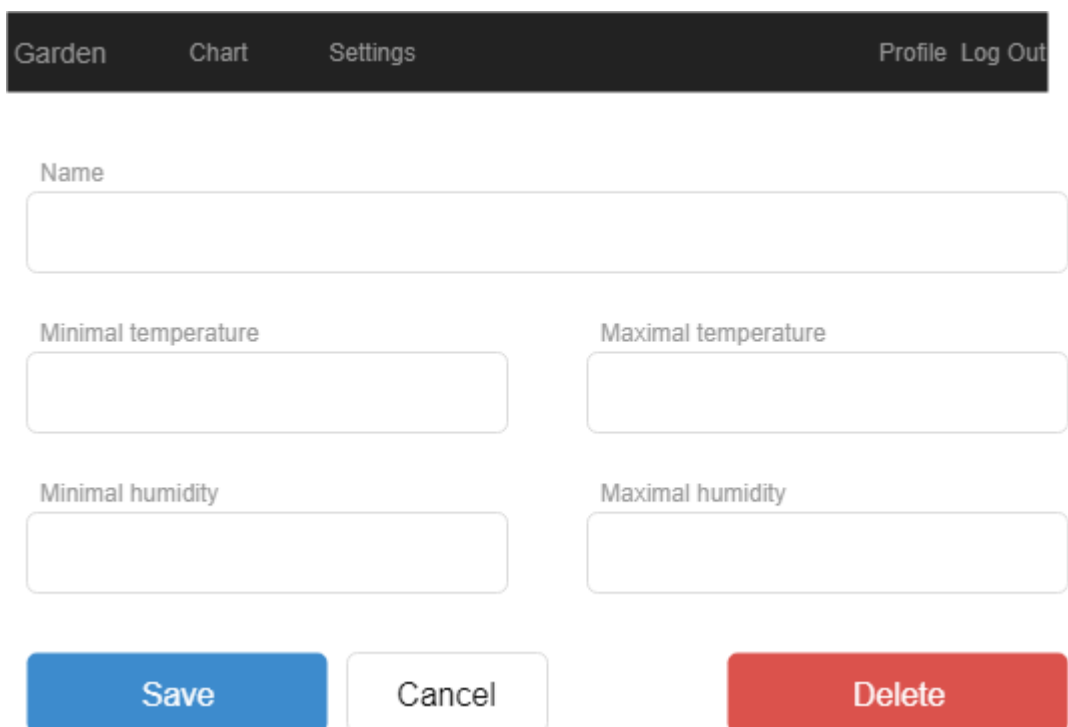
3.4.1. Menu



Obrázek 1 Menu

Navigační lišta obsahuje odkazy na úvodní obrazovku pro sledování právě naměřených hodnot květináčů, grafy naměřených hodnot, nastavení a odhlášení. Hlavní částí aplikace je seznam všech květináčů, které lze upravit a pozorovat aktuální hodnoty.

3.4.2. Nastavení

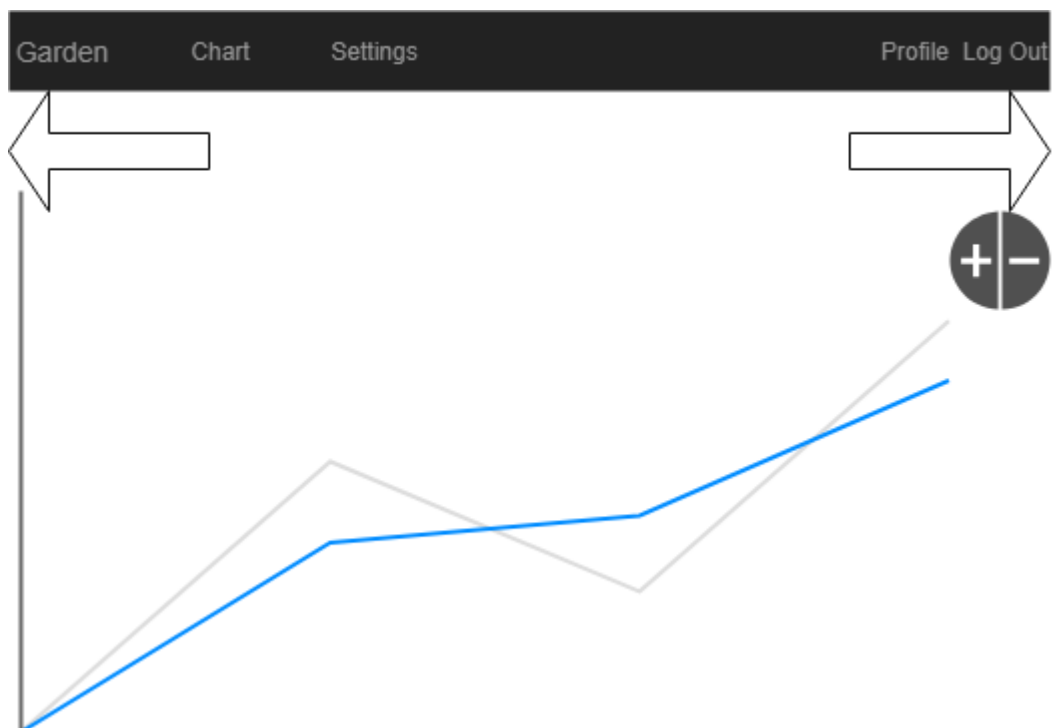


The screenshot shows the 'Settings' page of a web application. At the top, a dark navigation bar contains the links 'Garden', 'Chart', 'Settings', 'Profile', and 'Log Out'. Below this, the 'Name' field is a single-line text input. Underneath are four input fields arranged in a 2x2 grid: 'Minimal temperature', 'Maximal temperature', 'Minimal humidity', and 'Maximal humidity'. At the bottom of the form are three buttons: a blue 'Save' button, a white 'Cancel' button, and a red 'Delete' button.

Obrázek 2 Nastavení

Nastavení umožní uživateli změnit atributy květináče nebo smazat květináč. Uživatel bude mít možnost květináč pojmenovat a zvolit intervaly hodnot, které odpovídají ideálnímu prostředí pro jeho rostlinu. Tyto hodnoty budou zobrazeny v grafu měření.

3.4.3. Výsledky měření



Obrázek 3 Prezentace dat

Uživatel bude moci procházet výsledky dlouhodobého měření zvoleného květináče. Graf se zobrazí po úsecích s možností změny zobrazeného časového intervalu. Mezi daty budou znázorněny přednastavené minimální a maximální hodnoty pro kontrolu vhodného prostředí.

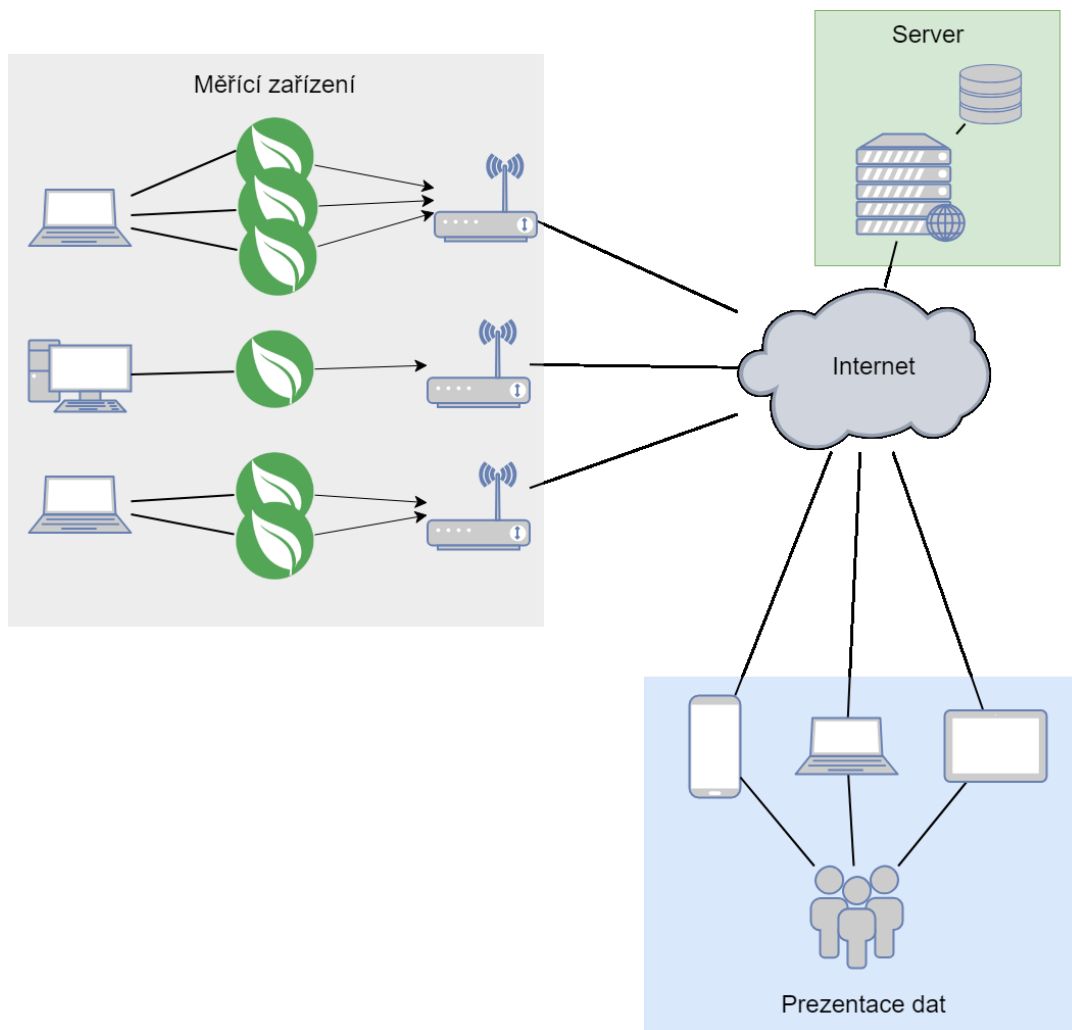
4. Analýza řešení

Celý projekt je rozdělen na několik částí. Výslednou strukturu projektu popisuje obrázek 4. První částí je měřicí zařízení, které bude pomocí senzorů měřit prostředí a posílat naměřená data přes internet pro další zpracování. Pro toto zařízení je potřeba vybrat vhodný hardware a implementovat software pro jeho ovládání, a aplikaci pro konfiguraci připojení k WiFi síti.

Uživatel bude pozorovat naměřená data odkudkoli prostřednictvím Internetu. Protože měřicí zařízení budou typicky připojeny k privátní síti, schované za překladem adres, data se odesílají na server, kde dojde k jejich zpracování pro další použití. Aplikace na serveru pracuje s databází, která umožní dlouhodobé uložení dat. Server musí zajistit autorizaci a autentizaci, aby nedošlo k neoprávněné manipulaci s uživatelskými daty.

Dále je nutné navrhnout vhodnou formu prezentace naměřených dat a způsob práce s více květináči. První možností by mohla být samostatná aplikace, která se připojí k serveru s rozhraním pro práci s květináči. Toto řešení by způsobilo nutnost zaměření se na konkrétní platformu. Lepší řešení má podobu webové aplikace, díky čemuž aplikace půjde spustit na každém zařízení s podporovaným webovým prohlížečem. Pro správné zobrazení i na mobilních zařízeních je nezbytné dbát na responsivní design.

Pro bližší popis webové aplikace je nutné si rozmyslet, jak bude vypadat typická komunikace serveru s klientem. Po přihlášení si uživatel vyžádá seznam svých květináčů a následně s nimi bude různě manipulovat a pravidelně se dotazovat na jejich aktuální naměřené hodnoty. Dále bude požadovat data za delší časové období konkrétního květináče. Zřejmě má komunikace převážně charakter požadavků na data potřebná pro vizualizace nebo žádostí o změnu dat na serveru. Takovou aplikaci je vhodné implementovat jako Single Page Application, čímž přesuneme logiku aplikace na klienta a server bude poskytovat rozhraní pro manipulaci s daty. Tím docílíme snížení zátěže na serveru a snížení množství přenášených dat. Zároveň bude práce v aplikaci plynulejší.



Obrázek 4: Struktura projektu

5. Výběr hardwaru

V současné době se na trhu nachází velké množství cenově dostupných mikropočítačů. Mezi velmi populární patří Raspberry PI. Jedná se o mikropočítač s moderním vícejádrovým procesorem architektury ARM. Zařízení, díky integrovanému WiFi čipu, velkému množství SDRAM paměti v řádech jednotek GB, USB a HDMI konektorům a podpory Linuxových operačních systémů, představuje plnohodnotný počítač [3]. Jeho cena, bez dodatečných senzorů, se pohybuje za hranicí 1000 Kč. Existují i méně vybavené a výkonné verze s cenou pod 400 Kč [4].

Na druhé straně se nachází elektronická stavebnice Arduino [5]. Oproti Raspberry PI nabízí možnost snadného vytváření vlastních prototypů. Cílí především na amatérské nadšence, kterým umožní navrhnout chytré zařízení bez hlubších informatických znalostí. Toto zařízení disponuje značnými nevýhodami. Mezi hlavní problémy spadá použitý mikrokontrolér od společnosti Atmel. Jedná se o jednoduchý čip pracující na nízké frekvenci v řádech malých desítek MHz s operační pamětí o velikosti několik KB a pamětí pro kód pouze v jednotkách KB. Cena základního počítače se pohybuje pod 1000 Kč, ale jeho výkon by mohl být pro budoucí rozšíření značně limitující.

Pro účely tohoto projektu byl vybrán mikropočítač STM32F4DISCOVERY, založený na architektuře ARM. Oproti Arduino nabízí mnohonásobně vyšší výkon a více paměti za cenu pohybující se pod 400 Kč [6]. Další velkou výhodou tohoto zařízení je jeho nízká spotřeba elektrické energie, která dosahuje hodnot kolem 40 mA [7].

Mikropočítač je připojen k rozšiřující desce STM32F4 Discovery Shield [8], do které lze snadno připojit rozšiřující moduly. Zároveň obsahuje integrovaný USB-UART modul pro sériovou komunikaci přes USB kabel. Pro komunikaci přes WiFi byl použit modul WiFi PLUS click [9]. Modul je kompatibilní se sítěmi podle standardu IEEE 802.11 b/g/n, podporuje zabezpečení typu WEP, WPA a WPA2 a obsahuje pomocný čip, který poskytuje rozhraní pro ovládání modulu prostřednictvím sériové komunikace UART. Dále byl použit modul SHT1x click

[10] se senzorem teploty a vlhkosti. Pro účely této práce budou použity pouze tyto senzory, nicméně nic by nemělo bránit budoucímu rozšíření o další.

6. Technologie

6.1. Měřicí zařízení

Firmware zařízení je napsán v jazyce C++, který oproti jazyku C nabízí pohodlnější práci, ale stále umožňuje snadné použití HAL knihovny [11] od výrobce.

6.2. Konfigurační aplikace

Konfigurační aplikace je napsána v jazyce C# s frameworkem .NET. Byla použita pouze jeho přenositelná část a celý kód je pomocí Mono přenositelný na další platformy.

6.3. Webová aplikace na straně serveru

Serverová část webové aplikace je naprogramována v jazyce C# s frameworkem .NET, který na rozdíl od novějšího .NET Core obsahuje kompletní dokumentaci a ukázky použití. Nicméně za cenu omezení některých technologií jen na platformu Windows.

Základ aplikace tvoří rozhraní OWIN [12] v podobě implementace Katana [13], která umožňuje přijímat HTTP zprávy, implementuje OAuth2 [14] a nabízí self-hosting. Komunikaci s databází zprostředkovává Entity Framework [15], který se postará o mapování dat a poskytuje nezávislost nad použitou databází včetně nekomerčních řešení. Tyto výhody jsou vykoupeny snížením výkonu při komunikaci s databází. Pro interakci s uživateli poskytuje aplikace rozhraní prostřednictvím Web API [16].

6.4. Webová aplikace na straně klienta

Vzhledem ke komplikovanějšímu kódu běžícímu na klientovi je výhodné použít vhodný framework. Naším hlavním požadavkem je, aby podporoval Single Page Application. V projektu je proto použit framework Angular 5, který zpřehledňuje kód rozdělením View, Controller a Service. Na rozdíl od původního

frameworku Angular.js nabízí Angular 5 vyšší bezpečnost, rychlost a zápis kódu v TypeScriptu [17].

Pro generování grafu je použita knihovna Chart.js [18], která umožňuje responzivní velikost grafu, a díky rozšíření ng2-charts [19] ji lze snadno použít přímo v Angularu. Pro definování stylu aplikace je použita knihovna Bootstrap, která umožňuje snadné vytvoření responsivního designu pro správné zobrazení na mobilních zařízeních [20]. Pro modifikaci ovládacích prvků Bootstrapu je použito rozšíření ng-Bootstrap [21]. Layout vychází z volně dostupné šablony SB Admin [30].

7. Řešení

7.1. Firmware měřicího zařízení

Po zapnutí čeká měřicí zařízení na konfigurační zprávy přijaté přes USB UART. Při obdržení zpráv provede potřebné konfigurace a po připojení k internetu měří pomocí senzoru SHT1x teplotu a vlhkost, a předává je modulu WiFi Plus click k odeslání na server. Komunikace s oběma moduly je implementována synchronně.

Zprávy se odešlou pomocí protokolu UDP. Preferování UDP před TCP spolu nese několik výhod a nevýhod. Nevýhodou je nižší spolehlivost a možnost doručení zpráv v nesprávném pořadí. První problém nám díky četnosti zpráv příliš nevádí, protože měřené veličiny se během desítek vteřin budou měnit velmi málo. Pro sledování současného stavu nebo grafu s výsledky posledních dní nejsou rozdíly v rámci i jednotek minut znatelné. Navíc velikost přenášených dat je velmi malá a při obvyklé velikosti MTU nebude docházet ke fragmentaci IP datagramů, poškození celé zprávy bude záviset pouze na stavu jednoho fragmentu.

Aby se zamezilo kolísání hodnot způsobeného vlivem doručení zpráv ve špatném pořadí, je každá zpráva opatřena číslem zprávy. Na serveru dochází k ošetření resetování počítadla zpráv. Mírně obtížnější manipulací se zprávami docílíme odstranění režie k zajištění spolehlivosti.

7.2. Aplikace pro konfiguraci

Jedná se o konzolovou aplikaci, která zpracovává uživatelský vstup z klávesnice a pomocí sériového portu COM komunikuje s měřicím zařízením prostřednictvím navrženého komunikačního protokolu.

7.3. Serverová část webové aplikace

Úkolem programu SPP.exe je přijímat naměřené hodnoty, spravovat databázi a poskytovat rozhraní pro klienta. Pokud není vstupními argumenty dáno jinak, je komunikace s klientem zabezpečena pomocí HTTPS.

Během konfigurace rozhraní OWIN jsou spuštěna vlákna pro příjem naměřených hodnot a uvolnění nevyužívaných květináčů z paměti každé dvě hodiny. Vzhledem k tomu, že webová aplikace si vytváří vlastní proces, by jinak bylo nutné implementovat komunikaci mezi procesy, což by způsobilo značnou režii navíc. Program může poskytovat veškerý obsah adresáře určeného pro klientský kód. Z bezpečnostních důvodů by adresář neměl obsahovat jakákoli citlivá data.

Pomocí techniky Code First definujeme model přímo v kódu aplikace. Podle něj jsou vytvořeny datové struktury v databázi, o což se postará Entity Framework.

Pro grafové zobrazení do databáze ukládáme měření reprezentující půlhodinový úsek. Pro sledování posledních naměřených hodnot a jejich zpracování se v paměti přechovává tabulka reprezentující aktivní květináče. Květináče v aplikaci představují jednotlivá měřicí zařízení, od kterých přichází data. Toto řešení velmi snižuje tlak na databázi, naopak ale zvyšuje paměťové nároky.

Pojmem aktivní květináč je myšlen květináč v aplikaci, který je připraven přijímat data. Květináč se aktivuje přiřazením párovacího řetězce. Aby se zamezilo přebytné spotřebě paměti v případě, kdy uživatel explicitně nedeaktivuje nevyužívaný květináč, pravidelně se každé dvě hodiny procházejí všechny aktivní květináče v tabulce a uvolňují ty, které nepřijaly od minulé kontroly data. Odstraněné květináče se nedeaktivují a v případě potřeby se znovu nahrají z databáze.

Aplikace je naprogramovaná jako Single Page Application, čímž oproti návrhovému vzoru Model View Controller snižuje tlak na server a síť, pomocí přesunu logiky klienta. Autentizace je dosaženo pomocí tokenů s omezenou životností. WebAPI zajišťuje registraci nových uživatelů, modifikaci květináčů, jejich přidávání a odebírání, nastavení párovacího řetězce a získání aktuálních hodnot nebo výsledků měření za požadované období.

7.4. Klientská část webové aplikace

Logika aplikace je rozprostřena mezi komponentami. Každá komponenta obsahuje vlastní view. Styl je definovaný globálně pomocí technologie Bootstrap. Pro předávání informací mezi komponentami a posílání dotazů na server se používají služby (services).

Při vstupu na stránku aplikace dojde k přesměrování nepřihlášených uživatelů na přihlašovací obrazovku. Přihlášení uživatelé jsou přesměrováni na obrazovku Garden. Přihlašovací obrazovka obsahuje mimo přihlašovacího formuláře i odkaz na registrační obrazovku, obsahující formulář pro registraci pomocí jména a hesla.

Garden obrazovka zobrazuje tabulku uživatelových květináčů se stránkováním po deseti a tlačítkem pro přidání. Na této obrazovce dochází k opakovanému procházení aktivních květináčů z aktuální stránky tabulky a pomocí asynchronního dotazu dochází k přepisování jejich posledních naměřených hodnot. Dále zde lze zvolit květináč kliknutím na jeho řádek v tabulce. Tím se zobrazí v navigační liště možnosti pro nastavení a zobrazení grafů.

8. Detaily implementace

8.1. Firmware měřicího zařízení

8.1.1. Komunikace se senzorem SHT1x

Rozhraní senzoru SHT1x se podobá komunikaci po I2C sběrnici, ale jeho komunikační protokol není kompatibilní s I2C protokolem, proto pro něj nelze použít standardní knihovny a je nutné přímo naimplementovat generování hodinového signálu (SCK) a datového signálu (DATA) dle datasheetu senzoru. K měření se používá přesnost výchozího nastavení: 14bit teplota a 12bit vlhkost. Pro zabránění nadměrnému zahřívání senzoru výrobce doporučuje držet senzor aktivní 10 % času [22]. Senzor po naměření dat začne vysílat signál. O naměřená data lze zažádat do startu nového měření. Vzhledem k délce měření až 80/320 ms při 12/14bit přesnosti následuje po měření každé veličiny pauza dvě sekundy. Tento čas se může v budoucnu využít pro komunikaci s dalšími senzory.

8.1.2. Komunikace s WiFi Plus click

Modul obsahuje čip MCW1001A [23], který pomocí komunikace UART nabízí rozhraní pro ovládání modulu. Z protokolu byly implementovány jen příkazy potřebné pro tento projekt. Modul poskytuje transportní protokoly TCP a UDP. Aplikační protokoly žádné naimplementuje.

Firmware implementuje připojení k přístupovým bodům se zabezpečením WEP, WPA a WPA2, nebo bez zabezpečení. Pro přenos dat na server je použit jednoduchý protokol, který umožňuje budoucí rozšíření o další senzory, se zachováním zpětné kompatibility.

8.1.3. Komunikační protokol mezi měřicím zařízením a serverem

Byte index	Popis
0:3	Číslo zprávy
4:11	ID květináče
12	Délka párovacího řetězce
13:N	Párovací řetězec
N+1	Počet senzorů
N+2+i3L	ID senzoru: 1: teplota 2: vlhkost vzduchu
N+2+i3L+1:	Naměřená data (BE), pro teplotu a vlhkost odpovídá stonásobku skutečné hodnoty
N+2+i3L+2	

Zpráva v korektním formátu obsahuje číslo zprávy pro detekci novějších dat, ID květináče z webové aplikace, délku párovacího řetězce a jeho data pro eliminaci špatně doručených dat při chybně zadaném ID. Následuje počet senzorů a jejich data s kódem příslušného senzoru. Data se ukládají jako celé číslo ve formátu Big Endian. Přenášené hodnoty teploty a vlhkosti se rovnají stonásobku skutečných hodnot.

8.1.4. Komunikační protokol mezi konfigurační aplikací a měřicím zařízením

Konfigurační aplikace posílá příkazy. Měřicí zařízení na příkazy odpovídá potvrzením (ACK) nebo zprávou s dodatečnými informacemi. Každá zpráva obsahuje hlavičku s typem zprávy v prvním bytu a délku přiložených dat v druhém bytu. Pro účely tohoto projektu není komunikační protokol zabezpečený a při odposlouchávání sériového portu by mohlo dojít k odhalení hesla přístupového bodu.

SCAN_REQ

Spustí v měřicím zařízení skenování dostupných přístupových bodů.

Byte index	Popis
0	Kód zprávy: 1
1	Délka: 0

Odpověď: SCAN_REPLY

SCAN_REPLY

Odpovídá na SCAN_REQ a předává počet výsledků.

Byte index	Popis
0	Kód zprávy: 2
1	Délka: 1
2	Počet nalezených přístupových bodů

SCAN_RES_REQ

Tato zpráva značí žádost o zaslání informací o i-tém přístupovém bodu. Číslování začíná nulou.

Byte index	Popis
0	Kód zprávy: 3
1	Délka: 1
2	Pořadí přístupového bodu

Odpověď: SCAN_RES_REPLY

SCAN_RES_REPLY

Zpráva obsahuje SSID a konfiguraci přístupového body podle poslední zprávy SCAN_RES_REQ. Konfigurační byte je shodný s konfigurací v komunikaci s MCW1001A [23] (kapitola 6.7.3).

Byte index	Popis
0	Kód zprávy: 4
1	Délka: 34
2	Délka SSID
3:34	SSID
35	Konfigurace přístupového bodu

SET_AP

Nastaví číslo přístupového bodu, ke kterému se bude měřicí zařízení připojovat.

Byte index	Popis
0	Kód zprávy: 5
1	Délka: 1
2	Pořadí přístupového bodu

Odpověď: ACK

SET_PASSWORD

Nastaví heslo pro připojení k přístupovému bodu.

Byte index	Popis
0	Kód zprávy: 10
1	Délka: N
2:N+1	Heslo

Odpověď: ACK

SET_REGION

Nastaví region pro dodržení standardu pro WiFi komunikaci. V Evropě se používá ETSI.

Byte index	Popis
0	Kód zprávy: 14
1	Délka: 1
2	Kód regionu: 0: FCC 1: IC 2: ETSI 3: SPAIN 4: FRANCE 5: JAPANA 6: JAPANB

Odpověď: ACK

SET_ID

Nastaví ID květináče pro spárování s květináčem ve webové aplikaci.

Byte index	Popis
0	Kód zprávy: 15
1	Délka: 8
2:9	ID květináče

Odpověď: ACK

CONNECT

Spustí připojování k nastavenému přístupovému bodu.

Byte index	Popis
0	Kód zprávy: 238
1	Délka: 0

Odpověď: CONNECT_REPLY

CONNECT_REPLY

Výsledek připojení je stejný s osmým bytem ekvivalentní zprávy v komunikaci s MCW1001A [23].

Byte index	Popis
0	Kód zprávy: 239
1	Délka: 2
2:3	Výsledek

CONNECTION_TYPE

Definuje nastavení připojení pomocí DHCP nebo statické.

Byte index	Popis
0	Kód zprávy: 32
1	Délka: 1
2	Způsob připojení: 0: DHCP 1: static

Odpověď: ACK

Statické nastavení

Příkaz pro statickou konfiguraci sítě. Obsahuje 4 B dat podle zprávy s příslušným kódem. IP adresa: 0x21, výchozí brána: 0x22, maska sítě: 0x23.

Byte index	Popis
0	Kód zprávy
1	Délka: 4
2:5	Data

Odpověď: ACK

ACK

Potvrzuje předchozí komunikaci. V případě potřeby signalizuje nastalé události.

Byte index	Popis
0	Kód zprávy: 0xFF
1	Délka: 1
2	Stav 0: OK 1: EVENT-následuje zpráva EVENT s dalšími informacemi 2: ERROR-následuje zpráva ERROR s dalšími informacemi

EVENT

Signalizuje změnu ve stavu připojení. Formát datové části odpovídá sedmému a osmému bytu v datasheetu MCW1001A [23] (odstavec 6.12.2.2).

Byte index	Popis
0	Kód zprávy: 0xFE
1	Délka: 2
2	Stav
3	Detail

ERROR

Signalizuje vážnou chybu v komunikaci. Před novou konfigurací je vyžadováno vypnutí a zapnutí měřicího zařízení. Formát datové části odpovídá sedmému a osmému bytu v datasheetu MCW1001A [23] (odstavec 6.12.2.5).

Byte index	Popis
0	Kód zprávy: 0xFE
1	Délka: 4
2:3	Data

8.2. Webová aplikace na straně serveru

8.2.1. MeasureCatcher

Poskytuje rozhraní pro přijímání zpráv a pravidelné uvolňování aktivních květináčů z paměti. Obě metody jsou určeny pro spuštění v samostatném vlákně po celou dobu běhu programu. Ke zprávě v korektním formátu se přidá čas doručení a v SyncGarden se hledá příslušný květináč. Při neúspěchu se kontroluje, zda nebyl uvolněn z paměti a případně se přidá zpět do SyncGarden. Nakonec se porovnají párovací řetězce a při shodě se data předají květináči.

8.2.2. SyncGarden

Implementuje rozhraní `IDictionary<long, ActivePot>` pro hledání květináčů podle ID. Jedná se o wrapper nad `Dictionary<long, ActivePot>` pro zajištění thread-safe vlastnosti všech operací kromě iterátorů. Častěji se přistupuje k datům pouze pro čtení, z tohoto důvodu se uvnitř třídy používá `ReadWriteLockSlim` pro efektivní zamykání.

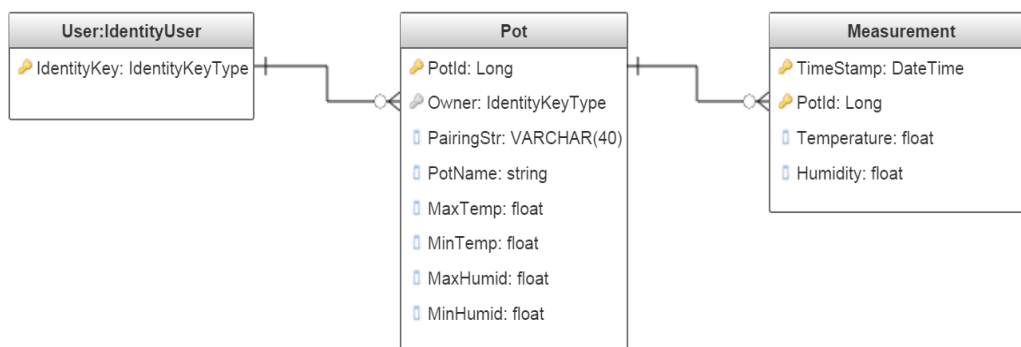
8.2.3. ActivePot

Reprezentuje aktivní květináč, který přijímá data. Podle pořadí odeslání se použije vždy jen novější zpráva. Zároveň se detekuje restartování čítače pořadí. Přijatá data si pamatuje, dokud nedostane zprávu, která je aspoň o půl hodiny novější než jeho nejstarší záznam. Poté uloží medián z dat senzorů do databáze. Zároveň poskytuje poslední přijatá data. Rozhraní třídy je thread-safe.

8.2.4. Model

Díky technice Code First se výsledné tabulky vygenerují podle popsaného modelu. Entity Framework umožňuje pro snadnější práci definovat položky pro obousměrnou referenci mezi tabulkami. Tyto položky se při generování tabulek

nepoužívají. Kódem popsaný model vygeneruje tabulky popsané obrázkem 2. Výsledný typ položek bude záviset na konkrétní použité databázi.



Obrázek 5 Model

Tabulka User je potomkem třídy IdentityUser z AspIdentity [24] s klíčem zděděným od předka. Každý User může vlastnit libovolný počet instancí typu Pot. Každý Pot má právě jednoho vlastníka typu User. Ekvivalentní relace se nachází mezi tabulkami Pot a Measurement. IdentityUser obsahuje několik datových položek, které se v projektu nepoužijí, jejich využití dává smysl pro budoucí rozšíření. Mezi přebytečné položky patří například e-mailová adresa a telefonní číslo.

8.2.5. WebAPI

Pro efektivní práci s databází využívají kontroléry webového rozhraní asynchronní komunikaci s databází. Rozhraní pro práci s květináči definují třídy GardenController a ChartController. Obě třídy jsou potomky abstraktní třídy PotController, která definuje metody pro kontrolu vlastnictví požadovaného květináče. GardenController poskytuje rozhraní pro modifikaci, vytvoření, smazání, aktivování a deaktivování květináče, získání naposledy naměřených dat aktivního květináče a seznamu květináčů. GraphController obsahuje metody pro získání naměřených hodnot teploty a vlhkosti vzduchu, za časový úsek dle parametrů data posledního požadovaného měření a počet dní zpět. AuthController zpracovává registrace nových uživatelů.

8.3. Klientský kód webové aplikace

8.3.1. AppModule

AppModule je hlavní modul aplikace. Obsahuje deklaraci komponent, importování použitých modulů a definici cest pro směrování v rámci aplikace.

8.3.2. AppComponent

Jedná se o vstupní bod aplikace. Nepřihlášené uživatele přesměruje na LoginController a přihlášené na GardenController. Dále definuje podmínky pro zobrazení položek navigační lišty. View této komponenty obsahuje navigační lištu. AppComponent je jediná komponenta, jejíž view zůstává zobrazený po celou dobu běhu aplikace.

8.3.3. LoginComponent a RegisterComponent

Obsahují formulář pro přihlášení a registraci pomocí služby AuthService a validují formuláře.

8.3.4. GardenComponent

Obsahuje tabulku květináčů. Zajišťuje načtení květináčů, jejich stránkování a výběr květináče.

8.3.5. SettingsModal

Komponenta, která používá Bootstrap Modal pro formulář nastavení a aktivaci květináče. Zprostředkovává nastavení, smazání, aktivaci/deaktivaci květináče a validaci formuláře. Při mazání květináče zobrazí ConfirmModal pro potvrzení.

8.3.6. ChartComponent

Zobrazuje graf pomocí modulu ng2-chart. Připravuje data pro graf pomocí funkce cacheOrReload. Pokud je požadovaný časový interval obsažen ve stažených datech, tak vybere požadovaná data. V opačném případě si od serveru vyžádá data

za celý interval. Dále komponenta obsahuje funkce pro výběr velikosti intervalu z hodnot 1, 7, 14, 21 a 30 dní, a posouván v čase o velikost intervalu.

8.3.7. PotService

Jedná se o službu, která zajišťuje veškerou komunikaci s rozhraním serveru pro práci s květináči. Všechny dotazy procházejí přes RequestInterceptor pro přidání hlavičky autentizace a ResponseInterceptor pro detekci prošlé platnosti tokenu. PotService spravuje seznam online květináčů na aktuální stránce a pomocí funkce lastValuesPoll dochází každé dvě sekundy k asynchronnímu dotazování na poslední naměřené hodnoty. Obsahuje factory metodu potFuncFactory pro vytváření dotazů se stejným obsahem, ale odlišným směrováním na rozhraní serveru.

9. Validace a zhodnocení

9.1. Testování

Veškerá komunikace měřicího zařízení byla ověřena měřením pomocí logického analyzátoru OMEGA. Funkce webové aplikace byly testovány při použití jednoho měřicího zařízení. Pro ověření práce s více uživateli byl navržen program pro generování požadavků na server.

V knihovně Chart.js byla objevena chyba v responzivní velikosti grafu, která způsobovala opakované zobrazování a mizení horizontálního posuvníku prohlížeče při některých velikostech okna. Tato chyba byla odstraněna fixním zobrazením posuvníku. Jiné řešení by bylo vypnutí zachování poměru stran grafu.

9.2. Obsluha více uživatelů

Pro simulaci více uživatelů byl naprogramován program Generator, pro jehož použití slouží inicializér StressTestDBInitializer. Inicializovaná databáze obsahuje 1000 uživatelů. Každý uživatel má přiřazeno 100 květináčů. Jeden květináč každého uživatele obsahuje vygenerovaná měření za jeden měsíc.

V prvním kroku testování aplikace dojde k přihlášení uživatelů a aktivaci všech květináčů. Následně se vyčlení jedno vlákno, které bude po zbytek testu generovat naměřená data pro všechny předpřipravené květináče.

Samotný test představuje opakované procházení všech uživatelů a paralelní posílání požadavků na server. Po odeslání požadavků všech uživatelů se vypíše hláška na konzoli, hlavní vlákno se na jednu sekundu uspí a následně se testuje znovu. Na server přicházejí požadavky na aktuální data pro deset květináčů, což odpovídá jedné zobrazené stránce v tabulce aplikace, požadavek pro data grafu za poslední měsíc, odstraňování, přidávání a načítání celého seznamu květináčů. Výběr následujícího požadavku se řídí pseudonáhodným generátorem. Jelikož nejčastějším požadavkem webového klienta je dotaz na nejnovější naměřené hodnoty, je výběr tohoto požadavku upřednostněn.

Testovací aplikace umožňuje zvolit počet uživatelů dle limitace vygenerovaných dat. Pro představu o chování aplikace byl test proveden pro 1000 aktivních uživatelů, přičemž ani po několikaminutovém běhu nedošlo vzhledem k vytížení serveru k odmítnutí zpracování jediného požadavku.

Vzhledem k povaze testovací aplikace, jakožto programu, který běží na jednom počítači, nelze automaticky předpokládat, že ke stejnému chování bude docházet i za běžného provozu. Přesto se lze domnívat, že výkon nebude zásadně nižší. V testovací aplikaci docházelo ke generování měřených dat neustále a uživatelé posílali data přibližně každé dvě sekundy. V projektu dochází k odesílání naměřených dat každé 4 sekundy, stejně jako u požadavků na poslední naměřené hodnoty. Zároveň klient neodesílá požadavky při každé uživatelské interakci, ale pouze v případě, že je nezbytné načíst další data.

Testování proběhlo na počítači se čtyřjádrovým procesorem Intel Core i5-7300HQ s 8 GB RAM.

9.3. Uživatelské rozhraní

9.3.1. Menu

Hlavním ovládacím prvkem je navigační lišta, která obsahuje odkazy na jednotlivé obrazovky a jméno uživatele. Při zobrazení v malých oknech se design navigační lišty změní na rozbalovací vertikální seznam, který je roztažen přes celou délku okna.

Po přihlášení je uživatel přesměrován na obrazovku Garden viz obrázek 6. Zde se nachází tabulka uživatelových květináčů a tlačítko pro přidání nového květináče. Pro přehlednost obsahuje tabulka stránky po deseti květináčích. Oproti původnímu návrhu této, je v tabulce zobrazena položka **State** označující stav květináče. Květináč je ve stavu offline, pokud nemá přiřazený párovací řetězec nebo měřící zařízení dlouho neodeslalo žádná data. Dále lze vybrat konkrétní květináč (kliknutím na jeho řádek) pro nastavení nebo zobrazení grafů.

Smart Garden

User0

Logout

Charts

Garden

Settings

Garden

Add

Name	ID	Temperature	Humidity	State
Flower1	1.0.0.0.0.0.0	22.35	11.43	online
Flower2	2.0.0.0.0.0.0	22.35	11.43	online
Flower3	3.0.0.0.0.0.0	22.35	11.43	online
Flower4	4.0.0.0.0.0.0	22.35	11.43	online
	5.0.0.0.0.0.0	22.35	11.43	online
	6.0.0.0.0.0.0	22.35	11.43	online
	7.0.0.0.0.0.0	22.35	11.43	online
	8.0.0.0.0.0.0	22.35	11.43	online
	9.0.0.0.0.0.0	22.35	11.43	online
	10.0.0.0.0.0.0	22.35	11.43	online
Name	ID	Temperature	Humidity	State

<

1

2

3

4

5

6

7

8

9

10

11

>

Obrázek 6 Menu

9.3.2. Nastavení

Obrazovka pro nastavení (viz obrázek 7) má podobu Bootstrap komponenty Modal, která je určena pro dialogová okna a zobrazuje se přes předchozí okno. Na rozdíl od návrhu zde přibyla položka pro nastavení aktivačního řetězce. Data z měřicího zařízení se párují s květináčem aplikace na základě shody ID a párovacího řetězce.

Settings [X]

Name: Flower2

ID: 2.0.0.0.0.0.0.0

Minimal temperature: 10

Maximal temperature: 25

Minimal humidity: 40

Maximal humidity: 60

Reset Delete Submit

Pairing string: pairingstring

Activated

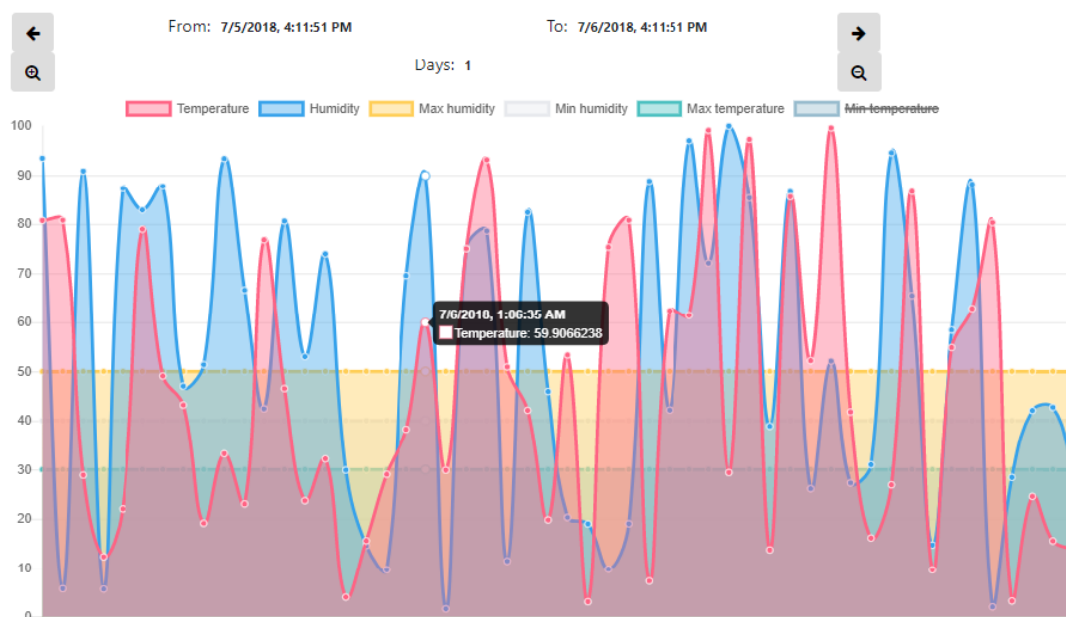
Close

Obrázek 7 Nastavení

9.3.3. Grafy

Obrazovka pro zobrazení grafů (viz obrázek 8) obsahuje ovládací prvky pro zobrazení grafu za větší nebo menší interval, a posun v čase o délku tohoto intervalu. Graf má pevně daný poměr stran a jeho velikost se mění podle velikosti okna prohlížeče.

Kliknutím na legendu grafu lze vypínat nebo zapínat zobrazení jednotlivých senzorů a uživatelem nastavených krajních hodnot cíleného intervalu hodnot. Jednotlivé grafy obsahují body. Pokud uživatel namíří kurzor na tyto body, zobrazí se popisek s informacemi o hodnotě a času měření.



Obrázek 8 Grafy

10. Související projekty

10.1. Parrot Pot

Projekt Parrot Pot funguje na způsob chytrého květináče, ke kterému se uživatel připojí mobilní aplikací pomocí Bluetooth. Nabízí systém automatického zavlažování a senzory na snímání intenzity světla, množství hnojiva, půdní vlhkost a množství vody v nádržce. Umožňuje výběr profilu pro více než osm tisíc rostlin. Nevýhodou tohoto řešení je možnost pracovat s květináči jen v dosahu Bluetooth a omezení pouze na mobilní systémy Android a iOS. [25]

10.2. Planty

Druhým projektem je Planty. Ten na rozdíl od předchozího projektu připojuje květináč k serveru a data z něj si lze prohlížet pomocí mobilní aplikace odkudkoli. Také nabízí velmi podobné funkce jako předchozí projekt. [26]

10.3. DIY

Další existující řešení představuje sada pro vlastní výrobu, kde lze ovládat senzory pomocí upravitelné stavebnice Arduino. Sada obsahuje jen senzory na měření teploty, vlhkosti a hladiny vody. Naměřené hodnoty lze prohlížet pomocí mobilní aplikace. Bohužel výrobce neposkytuje podrobnější informace o aplikaci. Dále projekt umožňuje přijímat notifikace pomocí e-mailu nebo sociální sítě Twitter. Nevýhodou tohoto řešení je náročnost sestavení a zapojení, nutnost zásahu koncového uživatele do zdrojového kódu pro připojení k jeho WiFi síti a následně pomocí nástrojů pro Arduino přeložit a nahrát kód. [27]

10.4. Porovnání

Na rozdíl od těchto projektů se tato práce zaměřuje na měření a práci s desítkami chytrých květináčů. Existující řešení poskytují pouze seznam květináčů a pro sledování aktuálních hodnot je u nich nutné přejít na obrazovku konkrétního květináče. Zároveň tento projekt má podobu webové aplikace, čímž se zbavuje závislosti na konkrétní platformě.

Během této práce byla největší pozornost věnována právě webové aplikaci. Z tohoto důvodu se tato práce omezuje jen na měření teploty a vlhkosti vzduchu. Dále by zde měla být lépe zabezpečena komunikace měřícího zařízení, případně změna konfigurační konzolové aplikace na aplikaci s GUI.

11. Závěr

Cílem práce bylo vytvořit chytrý květináč, zařízení pro měření prostředí rostliny. Výsledný projekt je tvořen ze tří částí. První částí je samotné zařízení pro měření dat. Pro toto zařízení byl vybrán vhodný hardware, a naimplementováno jeho ovládání a odesílání naměřených dat na server. Druhou částí je aplikace pro nakonfigurování měřícího zařízení přes sériovou linku. Třetí částí je webová aplikace, jejíž kód na straně serveru je postaven na rozhraní OWIN a klientský kód na frameworku Angular. Aplikace na straně serveru přijímá a zpracovává naměřené hodnoty, implementuje WebAPI rozhraní pro interakci s klientem a poskytuje klientský kód.

Klient obsahuje logiku webové aplikace a umožňuje uživateli zobrazovat grafy naměřených hodnot, sledovat aktuální data senzorů, modifikovat počet květináčů a jejich vlastnosti. Květináči lze přiřadit jeho jméno a minimální a maximální hodnoty senzorů, které se zobrazují v grafu spolu s naměřenými hodnotami a detekovat jejich překročení. Byla implementována jednoduchá autentizace a autorizace pomocí tokenů s omezenou životností. Pomocí technologie Bootstrap bylo docíleno responsivního designu a moderního stylu.

V serverové části webové aplikace je naimplementováno hostování aplikace pomocí OWIN, autorizace a autentifikace uživatelů, a rozhraní pro manipulaci s květináči a získání výsledků měření. Klientská část představuje logiku SPA postavené na frameworku Angular. Aplikace umožňuje nastavování květináčů, prohlížení současných hodnot a prohlížení dlouhodobého vývoje v grafu.

Budoucí rozšíření by mohlo zavést JWT [28], který by umožnil více možností spravování tokenů. Jedna z dalších úprav by mohlo být přidání senzorů světla a vodní hladiny. Zároveň se nabízí rozšířit zabezpečení celého projektu. Nahradiť komunikačního protokolu mezi měřícím zařízením a serverem za HTTPS. Autorizaci a autentizaci květináče implementovat pomocí jména a hesla, šifrovat komunikaci mezi měřícím zařízením a konfigurační aplikací. Také je důležité zamezit generování falešných požadavků přidávající data do databáze.

Seznam použité literatury

- [1] ROSE, Karen, ELDRIDGE, Scott, CHAPIN, Lyman, 2015. *THE INTERNET OF THINGS: AN OVERVIEW* [online]. ISOC [cit. 8.7.2018]. Dostupné z: <https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf>
- [2] Internet of Things, 2018. *Red Hat: We make open source technologies for the enterprise* [online]. Red Hat, Inc [cit. 1.7.2018]. Dostupné z: <https://www.redhat.com/en/insights/internet-of-things> (přeloženo autorem)
- [3] RASPBERRY PI 3 MODEL B+, 2018. *Raspberry Pi: Teach, Learn, and Make with Raspberry Pi* [online]. Pi Foundation [cit. 1.7.2018]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [4] RASPBERRY PI ZERO W, 2018. *Raspberry Pi: Teach, Learn, and Make with Raspberry Pi* [online]. Raspberry Pi Foundation [cit. 1.7.2018]. Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
- [5] Arduino Uno Rev3, 2018. *Arduino* [online]. Arduino [cit. 1.7.2018]. Dostupné z: <https://store.arduino.cc/arduino-uno-rev3>
- [6] STM32F4DISCOVERY, 2018. *Home: STMicroelectronics* [online]. STMicroelectronics [cit. 1.7.2018]. Dostupné z: <https://www.st.com/en/evaluation-tools/stm32f4discovery.html>
- [7] AN4365 Application note, 2014. *Home: STMicroelectronics* [online]. STMicroelectronics [cit. 1.7.2018]. Dostupné z: https://www.st.com/content/ccc/resource/technical/document/application_note/13/0a/06/b9/1e/2f/4d/9d/DM00096220.pdf/files/DM00096220.pdf/jcr:content/translations/en.DM00096220.pdf
- [8] STM32F4 Discovery Shield Manual, 2013. *MikroElektronika: Hardware and Software Tools for the Embedded World* [online]. MikroElektronika [cit. 1.7.2018]. Dostupné z: <https://download.mikroe.com/documents/add-on-boards/click-hields/stm32f4-discovery/stm32f4-discovery-mikrobus-shield-user-manual-v100.pdf>
- [9] WiFi Plus click, 2018. *MikroElektronika: Hardware and Software Tools for the Embedded World* [online]. MikroElektronika [cit. 1. 7. 2018]. Dostupné z: <https://www.mikroe.com/wifi-plus-click>

- [10] SHT1x click, 2018. *MikroElektronika: Hardware and Software Tools for the Embedded World* [online]. MikroElektronika [cit. 1. 7. 2018]. Dostupné z: <https://www.mikroe.com/sht1x-click>
- [11] UM1725 User Manual, 2013. *MikroElektronika: Hardware and Software Tools for the Embedded World* [online]. MikroElektronika [cit. 1. 7. 2018]. Dostupné z: https://www.st.com/content/ccc/resource/technical/document/user_manual/2f/71/ba/b8/75/54/47/cf/DM00105879.pdf/files/DM00105879.pdf/jcr:content/translations/en.DM00105879.pdf
- [12] OWIN, 2018. *OWIN* [online]. Ryan Riley, Scott Koon, Mark Rendle [cit. 1. 7. 2018]. Dostupné z: <http://owin.org/>
- [13] HOWARD, Dierking, 2013. An Overview of Project Katana, *Microsoft Docs* [online]. Microsoft [cit. 1. 7. 2018]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/aspnet/overview/owin-and-katana/an-overview-of-project-katana>
- [14] OWIN OAuth 2.0 Authorization Server, 2014. *Microsoft Docs* [online]. Hongye Sun, Praburaj Thiagarajan, Rick Anderson, Microsoft [cit. 1. 7. 2018]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/aspnet/overview/owin-and-katana/owin-oauth-20-authorization-server>
- [15] Entity Framework, 2013. *MSDN* [online]. Microsoft [cit. 1. 7. 2018]. Dostupné z: [https://msdn.microsoft.com/en-us/library/gg696172\(v=vs.103\).aspx](https://msdn.microsoft.com/en-us/library/gg696172(v=vs.103).aspx)
- [16] WASSON, Mike, 2017. Get Started with ASP.NET Web API 2. *Microsoft Docs* [online]. Microsoft [cit. 1. 7. 2018]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api>
- [17] Architecture overview, 2018. *Angular* [online] Google [cit. 1. 7. 2018]. Dostupné z: <https://angular.io/guide/architecture>
- [18] BRUNEL, Simon, 2018. Chart.js, *Chart.js* [online]. Chart.js Contributors [cit. 1. 7. 2018]. Dostupné z: <https://www.chartjs.org/>
- [19] ng2-charts, 2018. *Valar Software* [online]. Valor Software [cit. 1. 7. 2018]. Dostupné z: <https://valor-software.com/ng2-charts/>
- [20] Bootstrap, 2018. *Bootstrap* [online]. Twitter [cit. 1. 7. 2018]. Dostupné z: <https://getbootstrap.com/>

- [21] Angular powered Bootstrap, 2018. *Angular powered Bootstrap* [online] Angular ng-bootstrap team Software [cit. 1. 7. 2018]. Dostupné z: <https://ng-bootstrap.github.io/#/home>
- [22] Datasheet_SHT1x_V4.0_C1, 2008. *SparkFun Electronics* [online]. Sensirion [cit. 1. 7. 2018]. Dostupné z: https://www.sparkfun.com/datasheets/Sensors/SHT1x_datasheet.pdf
- [23] MCW1001A_DataSheet.book, 2011. *Home: Microchip Technology* [online]. Microchip Technology Inc. [cit. 1. 7. 2018]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70671A.pdf>
- [24] Introduction to ASP.NET Identity, 2013. *Microsoft Docs* [online]. Jon Galloway, Pranav Rastogi, Rick Anderson, Tom Dykstra, Microsoft [cit. 1. 7. 2018]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity>
- [25] Parrot POT, 2018 *Parrot* [online]. Parrot Drones SAS [cit. 1. 7. 2018]. Dostupné z: <https://www.parrot.com/us/connected-garden/parrot-pot#buy-a-parrot-pot-well-plant-a-tree->
- [26] Planty, 2018 *n.thing* [online]. N.THING [cit. 1. 7. 2018]. Dostupné z: <https://nthing.net/planty/>
- [27] Akin Yildiz, 2015. DIY SMART PLANT, *Instructables* [online]. Autodesk, Inc. [cit. 1. 7. 2018]. Dostupné z: <http://www.instructables.com/id/DIY-Smart-Plant-pot/>
- [28] RFC 7519, 2015. *IETF Tools* [online]. Michael B. Jones, John Bradley, Nat Sakimura [cit. 1. 7. 2018]. Dostupné z: <https://tools.ietf.org/html/rfc7519>
- [29] The Application Shell, 2018. *Angular* [online]. Google [cit. 1. 7. 2018]. Dostupné z: <https://angular.io/tutorial/toh-pt0>
- [30] MILLER, David, SB Admin, *Start Bootstrap* [online] Blackrock Digital [cit. 7. 7. 2018]. Dostupné z: <https://startbootstrap.com/template-overviews/sb-admin/>
- [31] GNU Arm Embedded Toolchain, 2018. ARM Developer [online]. ARM [cit. 15. 7. 2018]. Dostupné z: <https://developer.arm.com/open-source/gnu-toolchain/gnu-rm>
- [32] STM32CubeF4, 2018. Home – STMicroelectronics [online]. STMicroelectronics [cit. 15. 7. 2018.]. Dostupné z: <https://www.st.com/en/embedded-software/stm32cubef4.html>

Appendix 1 – kompilace a konfigurace

Měřicí zařízení

Pro správný překlad je nezbytné nainstalovat balík nástrojů s překladačem GNU Arm Embedded Toolchain [31] a knihovny od výrobce STM32CubeF4 [32]. Před přeložením je potřeba nastavit v souboru Makefile cestu k STM32CubeF4. Cesta k překladači by měla odpovídat výchozím cestám pro instalaci na Unixových operačních systémech. Překlad se spustí v příkazové řádce v adresáři projektu příkazem `make`. Po přeložení se kód nahraje do měřicího zařízení příkazem `make flash`. Volitelně lze použít pro ladění debugovací čip mikropočítače pomocí projektu OpenOCD.

Webová aplikace-server

Kompilace

Projekt obsahuje solution, který se přeloží pomocí programu Visual Studio.

Konfigurace

Aplikace, pro příjem zpráv z měřicího zařízení používá nestandardní protokol a poslouchá na portu 10000. Firewall některých antivirových programů může tuto komunikaci blokovat a je nutné v něm nastavit výjimku.

Současný uživatel musí mít povolení poslouchat na požadovanou URL. Toto nastavení lze provést v příkazové řádce spuštěné v administrátorském režimu a zadáním příkazu dle následující šablony: `netsh http add urlacl url=<celá URL> user=<uživatelské jméno>`. Pro použití aplikace ve výchozím nastavení stačí v příkazu zadat URL jako `https://+:443/`.

Nastavení certifikátu

Pro testovací účely lze vytvořit vlastní certifikát bez podpisu certifikační authority. V operačním systému Windows 10 lze takový certifikát vytvořit následujícím příkazem, v příkazové řádce spuštěné v režimu administrátora: `New-`

`SelfSignedCertificate -CertStoreLocation Cert:\LocalMachine\My -DnsName "localhost" -FriendlyName "SPPCert" -NotAfter (Get-Date).AddYears(10)`. Následně program vypíše hexadecimální číslo označené jako Thumbprint. Toto číslo bude použito v následujícím kroku.

V příkazovém řádku v režimu administrátora se propojí aplikace s certifikátem zadáním příkazu podle následující šablony: `netsh http add sslcert ipport=<IP:port> certhash=<thumbprint> appid={<GUID aplikace>}`. Pro zjednodušení lze zadat IP adresu jako 0.0.0.0. Tato adresa vyhovuje všem adresám lokálního stroje. Thumbprint je hexadecimální číslo a jedná se o SHA-1 hash certifikátu. GUID aplikace je číslo, které se vypíše na konzoli po zapnutí programu SPP.exe.

Inicializace

Pro testování lze použít inicializér databáze. K tomu slouží argument `initialize=<inicializér>`. Argument nastavte na `basic` pro deset uživatelů s deseti květináči. Uživatelé mají jméno ve tvaru `User<číslo uživatele>` s číslováním od nuly. Heslo je odvozeno podobně od šablony `password<číslo uživatele>`. Naměřená data bude obsahovat jen květináč s ID 1.0.0.0.0.0.0. (uživatele č. 0).

Nastavením argumentu na `big` se vygeneruje databáze s 1000 klienty se 100 květináčů pro každého uživatele. Každý uživatel má předpřipravené naměřené hodnoty za poslední měsíc pro květináč s nejnižším ID. Generování této databáze trvá déle, proto se používá především pro zátěžový test.

Webová aplikace-klient

Překlad aplikace se provede pomocí příkazové řádky prostředí Node.js s nainstalovaným Angular CLI. Instalace Angular CLI se provede příkazem `npm install -g @angular/cli`. Překlad se spustí příkazem `ng build` v adresáři projektu. Pro změnu adresáře slouží standardní příkaz `cd`. Přeložené soubory se nacházejí ve složce „dist“ a pro použití ve webové aplikaci je nutné je zkopírovat do složky „ready“. [29]

Appendix 2 – Uživatelská dokumentace

Spuštění měřicího zařízení

Po spuštění měřicího zařízení je nezbytné provést nastavení pomocí konfigurační aplikace. Uživatel připojí k měřicímu zařízení USB kabel do vstupu označeného jako USB UART a druhý konec připojí ke svému počítači. Následně spustí program `setter.exe`. Tento program pro své spuštění vyžaduje mít nainstalovaný .NET Framework ve verzi 4 nebo vyšší. Po spuštění programu se uživatel řídí zobrazovanými informacemi. V textovém vstupu, kromě SSID a hesla přístupového bodu, program nerozlišuje velká a malá písmena. Vstup se potvrzuje stisknutím klávesy ENTER. V případě neplatného vstupu dojde k výzvě k opětovnému zadání.

Nejprve je nutné vybrat sériový port COM, ke kterému je zařízení připojeno. V operačním systému Windows lze informaci nalézt ve Správci Zařízení. Následně uživatel zadá ID květináče, jehož prostředí chce měřit, z webové aplikace. Uživatel napíše IPv4 adresu serveru webové aplikace Smart Plant Pot a zadá řetězec pro párování. Jedná se o textový řetězec o délce deset až dvacet znaků, který se musí shodovat s párovacím řetězcem příslušného květináče ve webové aplikaci. Dále uživatel vybere region podle geografické polohy. Pro uživatele v Evropě je určen region ETSI. Následně dojde k vyhledání dostupných přístupových bodů (AP). Tato akce může trvat až 30 sekund. Po jejím dokončení program vypíše seznam názvů SSID dostupných AP a typ jejich zabezpečení. Zařízení podporuje zabezpečení WEP, WPA, WPA2. Následně uživatel vybere přístupový bod. Pokud je zabezpečený, bude vyzván k zadání hesla. Dále zvolí způsob nastavení sítě: DHCP pro automatické nebo STATIC pro statické nastavení. Pokud vybere statické nastavení bude vyzván k zadání IPv4 adresy pro měřicí zařízení, IPv4 adresy výchozí brány a masky sítě. Tyto údaje se zadávají v tečkové notaci, např. 192.168.1.1. Následně dojde k připojení měřicího zařízení k přístupovému bodu, které může trvat až 30 sekund. Po dokončení připojování se zobrazí zpráva o úspěšném připojení nebo oznámení o chybě. Při opětovné konfiguraci je doporučeno zařízení nejprve vypnout a zapnout.

Webová aplikace-server

Nastavení vlastní URL

Ve výchozím nastavení program poslouchá na URL `https://localhost:443`. Lze definovat vlastní URL s protokolem HTTP nebo HTTPS spuštěním programu s argumentem `url:<vaše URL>`. Při použití protokolu HTTPS je nezbytné nastavit certifikát.

Webová aplikace-klient

Přihlášení uživatele

Po vstupu na URL webové aplikace se nepřihlášenému uživateli zobrazí formulář pro přihlášení. Uživatel s existujícím profilem se přihlásí pomocí jména a hesla kliknutím na tlačítko Login pod formulářem. Při úspěšném přihlášení je uživatel přesměrován na okno Garden.

Registrace nového uživatele

Na konci přihlašovacího formuláře se nachází odkaz pro registraci. Po jeho rozkliknutí je uživatel přesměrován na registrační formulář. Pro registraci vyplní jméno, heslo a potvrzení hesla. Při nedodržení požadovaného formátu se zobrazí upozornění požadující změnou formátu. Pro odeslání registrace uživatel stiskne tlačítko Register. Při úspěšné registraci dojde k přesměrování na obrazovku pro přihlášení.

Základní práce s květináči

Přehled všech uživatelových květináčů se nachází v okně Garden. Do tohoto okna se uživatel dostane automaticky po přihlášení nebo kliknutím na odkaz s nápisem Garden v navigační liště. Květináče jsou uspořádány do tabulky se stránkami po deseti. Mezi stránkami lze přepínat tlačítky na konci tabulky. Tabulka obsahuje nepovinné jméno květináče, ID, aktuální hodnoty senzorů, pokud je ve stavu online, a stav květináče. Květináč je ve stavu offline, pokud nemá přiřazený párovací řetězec nebo poslední dvě hodiny od aktivování neodeslalo měřící zařízení žádná data. V opačném případě je květináč ve stavu online.

Nový květináč lze přidat kliknutím na tlačítko Add nad tabulkou.

Nastavení květináče

Pro nastavení květináče uživatel vybere květináč, který chce nastavit. Výběr provede kliknutím na řádek požadovaného květináče. Řádek vybraného květináče je označen šedou barvou. Po vybrání květináče se v navigační liště zobrazí odkaz s názvem Settings. Kliknutím na tento odkaz se vyvolá formulář s nastavením květináče. V tomto formuláři lze nastavit jeho jméno a krajní hodnoty senzorů pro detekci nevhodného prostředí v grafu. Změnu těchto atributů lze provést po kliknutí na tlačítko Submit. Dále může být květináč smazán kliknutím na tlačítko Delete, po kterém se zobrazí okno pro potvrzení požadavku na smazání. Ve spodní části nastavovacího formuláře se pak nachází kolonka pro vyplnění párovacího řetězce a tlačítko pro aktivaci nebo deaktivaci květináče.

Prohlížení grafů

Po výběru květináče uživatel klikne na odkaz s názvem Chart v navigační liště. Zobrazí se graf hodnot za poslední den. Nad grafem se nachází legenda s barevným označením všech přístupných grafů, zobrazujících vlhkost, teplotu a uživatelem nastavené hodnoty pro maximum a minimum těchto veličin. Po kliknutí na tyto položky lze vypínat a zapínat zobrazení jednotlivých grafů. Každý graf obsahuje kulaté body, které po přejetím kurzoru myši nebo kliknutím prstu na dotykové obrazovce, zobrazí naměřenou hodnotu a čas měření.

Nad grafem se dále nachází datum, čas zobrazeného intervalu a délka intervalu v podobě počtu zobrazených dní. Pomocí tlačítek se znakem šipky lze posunovat grafem v čase o velikost intervalu. Tlačítka ve tvaru lupy lze zvětšovat či zmenšovat interval. Velikost intervalu nabývá hodnot 1, 7, 14, 21 a 30 dní.

Apendix 3 – přiložené CD

Doprovodné CD obsahuje generovanou dokumentaci a zdrojový kód serverové a klientské části, konfigurační aplikace, firmwaru zařízení a zátěžového testu. Detailní popis se nachází v souboru Readme.txt. Pro překlad je nezbytné přidat knihovny dle tohoto textového souboru.